Claims

1. A scripting method for controlling process flow and configuration in a distributed object-oriented software system, the method comprising:

providing a plurality of software components at least one of which is derived from a common base class and arranged for operation in the distributed software system;

providing an executable general container process;

creating a configuration file readable by the general container process for configuring the container process and including therein identification of at least one of the said software components;

in the general container process, reading the configuration file at run time and loading the software components identified in the configuration file for subsequent execution; and

further providing at least one rule set associated with one of the identified software components derived from the common base class for controlling process flow during execution of the corresponding component, responsive to run time conditions, without having to modify the component source code.

2. A scripting method according to claim 1 wherein the rule set is included within the configuration file.

3. A scripting method according to claim 2 wherein the configuration file is expressed in XML syntax.

4. A scripting method according to claim 3 wherein the common base class includes a rules engine that implements a predetermined rules-based scripting language; the rules engine including methods for selecting and executing the rule set.

5. A scripting method according to claim 4 wherein the rules engine implements a callback function for calling a selected one of the loaded software components to perform a predetermined function.

6. A scripting method according to claim 4 wherein the rules engine is arranged to receive a reference and to pass the reference to a component in connection with a callback to identify a resource to the called component.

7. A scripting method according to claim 4 wherein the rules engine implements a process rule type which encapsulates a selected set of children rules into a group for calling such group by a single name.

8. A scripting method according to claim 4 wherein the rules engine implements a Component rule type which makes calls automatically against all components of a specified type.

9. A scripting method according to claim 4 wherein the rules engine implements a Variable rule type which allows a string value to be associated with a name.

10. A distributed, object-oriented software system for enabling a user to script its own specific business logic without modifying the system source code, the system comprising:

a plurality of software components at least one of which is derived from a common base class and arranged for operation in the distributed software system;

a general container process for loading and executing selected ones of the said software components; and

a configuration file readable by the general container process and including therein identification of at least one of the said software components for loading and execution;

wherein the common base class includes a rules engine that implements a predetermined rules-based scripting language; and

wherein the configuration file includes at least one rule set consistent with the scripting language and associated with one of the identified software components derived from the common base class for realizing the user's specific business logic during execution of the corresponding component without having to modify the component source code.

11.   A software system according to claim 10 wherein the scripting language implements at least one rule type selected from the following:

a RuleSet rule type to make a runtime selection among multiple rule sets within a single component;

an Instruction rule type that makes a callback into the associated component to perform a predetermined function;

a Resource rule type providing identification of data/argument to pass into a callback;

a Process rule type which encapsulates a selected set of children rules into a group, thereby enabling more than one of such rules to be called by a single name;

a Variable rule type which allows a string value to be associated with a name;

an Acquire rule type to facilitate obtaining a selected Resource for use in later rules processing;

a Release rule type to explicitly de-allocate memory previously allocated to a selected resource; and

a Component rule type which makes calls against components of a specified type.

12.   A rules based scripting language for use in a distributed object oriented software system, wherein a software component of the software system is adapted to interpret and implement the scripting language, comprising:

a RuleSet rule type that instructs the component to make a runtime selection among multiple rule sets within the component;

an Instruction rule type that instructs the component to make a callback into the component to perform a predetermined function;

a Resource rule type that provides the component with identification of data/argument to pass into a callback;

a Process rule type which instructs the component to handle a selected set of children rules as a group, thereby enabling more than one of such rules to be called by a single name;

a Variable rule type which instructs the component to allow a string value to be associated with a name;

an Acquire rule type which instructs the component to obtain a selected Resource for use in later rules processing;

5 a Release rule type which instructs the component to explicitly de-allocate memory previously allocated to a selected resource; and

a Component rule type which instructs the component to make calls against other components of a specified type.

10